

Learning Concept Embeddings for Efficient Bag-of-Concepts Densification

Walid Shalaby, Wlodek Zadrozny

Computer Science Department

UNC Charlotte

wshalaby, wzadrozny@uncc.edu

Abstract

Explicit concept space models have proven efficacy for text representation in many natural language and text mining applications. The idea is to embed textual structures into a semantic space of concepts which captures the main topics of these structures. That so called bag-of-concepts representation suffers from data sparsity causing low similarity scores between similar texts due to low concept overlap. In this paper we propose two neural embedding models in order to learn continuous concept vectors. Once learned, we propose an efficient vector aggregation method to generate fully dense bag-of-concepts representations. Empirical results on a benchmark dataset for measuring entity semantic relatedness show superior performance over other concept embedding models. In addition, by utilizing our efficient aggregation method, we demonstrate the effectiveness of the densified vector representation over the typical sparse representations for dataless classification where we can achieve at least same or better accuracy with much less dimensions.

Explicit concept space models are motivated by the idea that high level cognitive tasks such learning and reasoning are supported by the knowledge we acquire from concepts (Song et al., 2015). Therefore, such models utilize concept vectors (a.k.a bag-of-concepts (BOC)) as the underlying semantic representation of a given text through a process called conceptualization, which is mapping the text into relevant concepts capturing its main topics. The concept space typically include concepts obtained from KBs such as Wikipedia, Probase (Wu et al., 2012), and others. Once the concept vectors are generated, similarity between two concept vectors can be computed using a suitable similarity/distance measure such as cosine.

The BOC representation has proven efficacy for semantic analysis of textual data especially short texts where contextual information is missing or insufficient. For example, measuring semantic similarity/relatedness (Gabrilovich and Markovitch, 2007; Kim et al., 2013; Shalaby and Zadrozny, 2015), dataless classification (Chang et al., 2008; Song and Roth, 2014, 2015; Li et al., 2016), short text clustering (Song et al., 2015), search and relevancy ranking (Egozi et al., 2011), event detection and coreference resolution (Peng et al., 2016).

1 Introduction

Vector-based semantic mapping models are used to represent textual structures (words, phrases, and documents) as high-dimensional *meaning* vectors. Typically, these models utilize textual corpora and/or Knowledge Bases (KBs) to acquire world knowledge, which is then used to generate a vector representation for the given text in the semantic space. The goal is thus to accurately place semantically similar structures close to each other in that semantic space. On the other hand, dissimilar structures should be far apart.

Similar to the traditional bag-of-words representation, the BOC vector is a high dimensional sparse vector whose dimensionality is the same as the number of concepts in the employed KB (typically millions). Consequently, it suffers from data sparsity causing low similarity scores between similar texts due to low concept overlap. Formally, given a text snippet $T = \{t_1, t_2, \dots, t_n\}$ of n terms where $n \geq 1$, and a concept space $C = \{c_1, c_2, \dots, c_N\}$ of size N . The BOC vector $\mathbf{v} = \{w_1, w_2, \dots, w_s\} \in \mathbb{R}^N$ of T is a vector of weights of each concept where each w_i of concept

c_i is calculated as in equation 1:

$$w_i = \sum_{j=1}^n f(c_i, t_j), 1 \leq i \leq N \quad (1)$$

Here $f(c, t)$ is a scoring function which indicates the degree of association between term t and concept c . For example, [Gabrilovich and Markovitch \(2007\)](#) proposed Explicit Semantic Analysis (ESA) which uses Wikipedia articles as concepts and the TF-IDF score of the terms in these article as the association score. Another scoring function might be the co-occurrence count or Pearson correlation score between t and c . As we can notice, only very small subset of the concept space would have non-zero scores with the given terms. Moreover, the BOC vector is generated from the *topn* concepts which have relatively high association scores with the input terms (typically few hundreds). Thus each text snippet is mapped to a very sparse vector of millions of dimensions having only few hundreds non-zero values ([Peng et al., 2016](#)).

Typically, the cosine similarity measure is used compute the similarity between a pair of BOC vectors \mathbf{u} and \mathbf{v} . Because the concept vectors are very sparse, we can rewrite each vector as a vector of tuples (c_i, w_i) . Suppose that $\mathbf{u} = \{(c_{n_1}, u_1), \dots, (c_{n_{|\mathbf{u}|}}, u_{|\mathbf{u}|})\}$ and $\mathbf{v} = \{(c_{m_1}, v_1), \dots, (c_{m_{|\mathbf{v}|}}, v_{|\mathbf{v}|})\}$, where u_i and v_j are the corresponding weights of concepts c_{n_i} and c_{m_j} respectively. And n_i, m_j are the indices of these concepts in the concept space C such that $1 \leq n_i, m_j \leq N$. Then, the relatedness score can be written as in equation 2:

$$Sim_{cos}(\mathbf{u}, \mathbf{v}) = \frac{\sum_{i=1}^{|\mathbf{u}|} \sum_{j=1}^{|\mathbf{v}|} \mathbb{1}(n_i=m_j) u_i v_j}{\sqrt{\sum_{i=1}^{|\mathbf{u}|} u_i^2} \sqrt{\sum_{j=1}^{|\mathbf{v}|} v_j^2}} \quad (2)$$

where $\mathbb{1}$ is the indicator function which returns 1 if $n_i=m_j$ and 0 otherwise. Having such sparse representation and using exact match similarity scoring measure, we can expect that two very similar text snippets might have zero similarity score if they map to different but very related set of concepts ([Song and Roth, 2015](#)).

Neural embedding models have been proposed to overcome the BOC sparsity problem. The basic idea is to learn fixed size continuous vectors for each concept. These vectors can then be used to compute concept-concept similarity and thus overcome the concept mismatch problem.

In this paper we propose two neural embedding models in order to learn continuous concept vectors based on the skip-gram model ([Mikolov et al., 2013b](#)). Our first model is the *Concept Raw Context model* (CRC) which utilizes concept mentions in a large scale KB to jointly learn embeddings of both words and concepts. Our second model is the *Concept-Concept Context model* (3C) which learns the embeddings of concepts from their conceptual contexts (i.e., contexts containing surrounding concepts only). After learning the concept vectors, we propose an efficient concept vector aggregation method to generate fully dense BOC representations. Our efficient aggregation method allows measuring the similarity between pairs of BOC vectors in linear time. This is more efficient than prior methods which require quadratic time or at least log-linear time if optimized (see equation 2).

We evaluate our embedding models on two tasks:

1. An intrinsic task of measuring entity semantic relatedness where our CRC model outperforms other concept embedding models.
2. An extrinsic task of dataless classification. Experimental results show that we can achieve better accuracy using our efficient BOC densification method compared to the original BOC sparse representation.

The contributions of this paper are threefold: First, we propose two low cost concept embedding models which requires few hours rather than days to train. Second, we propose simple and efficient vector aggregation method to obtain fully densified BOC vectors in linear time. Third, we demonstrate through experiments that we can obtain same or better accuracy using the densified BOC representation with much less dimensions (few in most cases), reducing the computational cost of generating the BOC vector significantly.

2 Related Work

Concept/Entity Embeddings: neural embedding models have been proposed to learn distributed representations of concepts/entities. [Song and Roth \(2015\)](#) proposed using the popular Word2Vec model ([Mikolov et al., 2013a](#)) to obtain the embeddings of each concept by averaging the vectors of the concept’s individual words. For example, the embeddings of *Microsoft Office* would be obtained by averaging the embeddings of *Mi-*

crosoft and *Office* obtained from the Word2Vec model. Clearly, this method disregards the fact that the semantics of multi-word concepts is different from the semantics of their individual words.

More robust concept embeddings can be learned from the concept’s corresponding article and/or from the structure of the employed KB (e.g., its link graph). Such concept embedding models were proposed by [Hu et al. \(2015\)](#); [Li et al. \(2016\)](#); [Yamada et al. \(2016\)](#) who all utilize the skip-gram model ([Mikolov et al., 2013b](#)), but differ in how they define the context of the target concept.

[Li et al. \(2016\)](#) extended the embedding model proposed by [Hu et al. \(2015\)](#) by jointly learning concept and category embeddings from contexts defined by all other concepts in the target concept’s article as well as its category hierarchy in Wikipedia. This method has the advantage of learning embeddings of both concepts and categories simultaneously. However, defining the concept contexts as pairs of the target concept and all other concepts appearing in its corresponding article might introduce noisy contexts, especially for long articles. For example, the Wikipedia article for *United States* contains links to *Kindergarten*, *First grade*, and *Secondary school* under the Education section.

[Yamada et al. \(2016\)](#) proposed a method based on the skip-gram model to jointly learn embeddings of words and concepts using contexts generated from surrounding words of the target concept or word. The authors also proposed incorporating the KB link graph by generating contexts from all concepts with outgoing link to the target concept to better model concept-concept relatedness.

Unlike [Li et al. \(2016\)](#) and [Hu et al. \(2015\)](#) who learn concept embeddings only, our CRC model (described in Section 3), maps both words and concepts into the same semantic space. Therefore we can easily measure word-word, word-concept, and concept-concept semantic similarities. In addition, compared to [Yamada et al. \(2016\)](#) model, we utilize contexts generated from both surrounding words and concepts. Therefore, we can better capture local contextual information of each target word/concept. Moreover, our proposed models are computationally less costly than [Hu et al. \(2015\)](#) and [Yamada et al. \(2016\)](#) models as they require few hours rather than days to train on similar computing resources.

BOC Densification: distributed concept vectors have been used by BOC densification mechanisms to overcome the BOC sparsity problem. [Song and Roth \(2015\)](#) proposed three different mechanisms for aligning the concepts at different indices given a sparse BOC pair (\mathbf{u}, \mathbf{v}) in order to increase their similarity score.

The many-to-many mechanism works by averaging all pairwise similarities. The many-to-one mechanism works by aligning each concept in \mathbf{u} with the most similar concept in \mathbf{v} (i.e., its best match). Clearly, the complexity of these two mechanisms is quadratic. The third mechanism is the one-to-one. It utilizes the Hungarian method in order to find an optimal alignment on a one-to-one basis ([Papadimitriou and Steiglitz, 1982](#)). This mechanism performed the best on dataless classification and was also utilized by [Li et al. \(2016\)](#). However, the Hungarian method is a combinatorial optimization algorithm whose complexity is polynomial. Our proposed densification mechanism is more efficient than these three mechanisms as its complexity is linear with respect to the number of non-zero elements in the BOC vector. Additionally, it is simpler as it does not require tuning a cut off threshold for the minimum similarity between two aligned concepts as in prior work.

3 Concept Embeddings for BOC Densification

A main objective of learning concept embeddings is to overcome the inherent problem of data sparsity associated with the BOC representation. Here we try to learn continuous concept vectors by building upon the skip-gram embedding model ([Mikolov et al., 2013b](#)). In the conventional skip-gram model, a set of contexts are generated by sliding a context window of predefined size over sentences of a given text corpus. Vector representation of a target word is learned with the objective to maximize the ability of predicting surrounding words of that target word.

Formally, given a training corpus of V words w_1, w_2, \dots, w_V . The skip-gram model aims to maximize the average log probability:

$$\frac{1}{V} \sum_{i=1}^V \sum_{-s \leq j \leq s, j \neq 0} \log p(w_{i+j} | w_i) \quad (3)$$

where s is the context window size, w_i is the target word, and w_{i+j} is a surrounding context word.

The softmax function is used to estimate the probability $p(w_O|w_I)$ as follows:

$$p(w_O|w_I) = \frac{\exp(\mathbf{v}_{w_O}^\top \mathbf{u}_{w_I})}{\sum_{w=1}^W \exp(\mathbf{v}_w^\top \mathbf{u}_{w_I})} \quad (4)$$

where \mathbf{u}_w and \mathbf{v}_w are the input and output vectors respectively and W is the vocabulary size. Mikolov et al. (2013b) proposed hierarchical softmax and negative sampling as efficient alternatives to approximate the softmax function which becomes computationally intractable when W becomes huge.

Our approach genuinely learns distributed concept representations by generating concept contexts from mentions of those concepts in large encyclopedic KBs such as Wikipedia. Utilizing such annotated KBs eliminates the need to manually annotate concept mentions and thus comes at no cost.

3.1 Concept Raw Context Model (CRC)

In this model, we jointly learn the embeddings of both words and concepts. First, all concept mentions are identified in the given corpus. Second, contexts are generated for both words and concepts from both other surrounding words and other surrounding concepts as well. After generating all the contexts, we use the skip-gram model to jointly learn words and concepts embeddings. Formally, given a training corpus of V words w_1, w_2, \dots, w_V . We iterate over the corpus identifying words and concept mentions and thus generating a sequence of T tokens t_1, t_2, \dots, t_T where $T < V$ (as multi-word concepts will be counted as one token). Afterwards we train the a skip-gram model aiming to maximize:

$$\frac{1}{T} \sum_{i=1}^T \sum_{-s \leq j \leq s, j \neq 0} \log p(t_{i+j}|t_i) \quad (5)$$

where as in the conventional skip-gram model, s is the context window size. In this model, t_i is the target token which would be either a word or a concept mention, and t_{i+j} is a surrounding context word or concept mention.

This model is different from Yamada et al. (2016) anchor context model in three aspects: 1) while generating target concept contexts, we utilize not only surrounding words but other surrounding concepts as well, 2) our model aims to maximize $p(t_{i+j}|t_i)$ where t could be a word or a concept, while Yamada et al. (2016) model

maximizes $p(w_{i+j}|e_i)$ where e_i is the target concept/entity (see Yamada et al. (2016) Eq. 6), and 3) in case t_i is a concept, our model captures all the contexts in which it appeared, while Yamada et al. (2016) model generates for each entity one context of s previous and s next words. We hypothesize that considering both concepts and individual words in the optimization function would generate more robust embeddings.

3.2 Concept-Concept Context Model (3C)

Inspired by the distributional hypothesis (Harris, 1954), we, in this model, hypothesize that "similar concepts tend to appear in similar conceptual contexts". In order to test this hypothesis, we learn concept embeddings by training a skip-gram model on contexts generated solely from concept mentions. As in the CRC model, we start by identifying all concept mentions in the given corpus. Then, contexts are generated from only surrounding concepts. Formally, given a training corpus of V words w_1, w_2, \dots, w_V . We iterate over the corpus identifying concept mentions and thus generating a sequence of C concept tokens c_1, c_2, \dots, c_C where $C < V$. Afterwards we train the skip-gram model aiming to maximize:

$$\frac{1}{C} \sum_{i=1}^C \sum_{-s \leq j \leq s, j \neq 0} \log p(c_{i+j}|c_i) \quad (6)$$

where s is the context window size, c_i is the target concept, and c_{i+j} is a surrounding concept mention within s mentions.

This model is different from Li et al. (2016) and Hu et al. (2015) as they define the context of a target concept by all the other concepts which appear in the concept's corresponding article. Clearly, some of these concepts might be irrelevant especially for very long articles which cite hundreds of other concepts. Our 3C model, alternatively, learns concept semantics from surrounding concepts and not only from those that are cited in its article. We also extend the context window beyond pairs of concepts allowing more influence to other nearby concepts.

The main advantage of the 3C model over the CRC model is its computational efficiency where the model vocabulary is limited to the corpus concepts (Wikipedia in our case). On the other hand, the CRC model is advantageous because it jointly learns the embeddings of words and concepts and is therefore expected to generate higher

quality vectors. In other words, the CRC model can capture more contextual signals such as actions, times, and relationships at the expense of training computational cost.

3.3 Training

We utilize a recent Wikipedia dump of August 2016¹, which has about 7 million articles. We extract articles plain text discarding images and tables. We also discard *References* and *External links* sections (if any). We pruned both articles not under the main namespace and pruned all redirect pages as well. Eventually, our corpus contained about 5 million articles in total.

We preprocess each article replacing all its references to other Wikipedia articles with the their corresponding page IDs. In case any of the references is a title of a redirect page, we use the page ID of the original page to ensure that all concept mentions are normalized.

Following Mikolov et al. (2013b), we utilize negative sampling to approximate the softmax function by replacing every $\log p(w_O|w_I)$ term in the softmax function (equation 4) with:

$$\log \sigma(\mathbf{v}_{w_O}^T \mathbf{u}_{w_I}) + \sum_{s=1}^k \mathbb{E}_{w_s \sim P_n(w)} [\log \sigma(-\mathbf{v}_{w_s}^T \mathbf{u}_{w_I})] \quad (7)$$

where k is the number of negative samples drawn for each word and $\sigma(x)$ is the sigmoid function ($\frac{1}{1+e^{-x}}$). In the case of the CRC model w_I and w_O would be replaced with t_i and t_{i+j} respectively. And in the case of the 3C model w_I and w_O would be replaced with c_i and c_{i+j} respectively.

For both the CRC & 3C models with use a context window of size 9 and a vector of 500 dimensions. We train the skip-gram model for 10 iterations using 12 cores machine with 64GB of RAM. The CRC model took ~ 15 hours to train for a total of ~ 12.7 million tokens. The 3C model took ~ 1.5 hours to train for a total of ~ 4.5 million concepts.

3.4 BOC Densification

As we mentioned in the related work section, the current mechanisms for BOC densification are inefficient as their complexity is least quadratic with respect to the number of non-zero elements in the BOC vector. Here, we propose simple and efficient vector aggregation method to obtain fully

densified BOC vectors in linear time. Our mechanism works by performing a weighted average of the embedding vectors of all concepts in the given BOC. This operation scales linearly with the number of non-zero dimensions in the BOC vector. In addition, it produces a fully dense vector representing the semantics of the original concepts and considering their weights. Formally, given a sparse BOC vector $\mathbf{s} = \{(c_1, w_1), \dots, (c_{|s|}, w_{|s|})\}$ where w_i is weight of concept c_i . We can obtain the dense representation of \mathbf{s} as in equation 8:

$$\mathbf{s}_{dense} = \frac{\sum_{i=1}^{|s|} w_i \cdot \mathbf{u}_{c_i}}{\sum_{i=1}^{|s|} w_i} \quad (8)$$

where \mathbf{u}_{c_i} is the embedding vector of concept c_i . Once we have this dense BOC vector, we can apply the cosine measure to compute the similarity between a pair of dense BOC vectors.

As we can notice, this weighted average is done once and for all for a given BOC vector. Other mechanisms that rely on concept alignment (Song and Roth, 2015), require realignment every time a given BOC vector is compared to another BOC vector. Our approach improves the efficiency especially in the context of dataless classification with large number of classes. Using our densification mechanism, we apply the weighted average for each class vector and for each instance once.

Interestingly, our densification mechanism allows us to densify the sparse BOC vector using only the top few dimensions. As we will show in the experiments section, we can get (near-)best results using these few dimensions compared to densifying with all the dimensions in the original sparse vector. This property reduces the cost of obtaining a BOC vector with a few hundred dimensions in the first place.

4 Experiments

4.1 Entity Semantic Relatedness

We evaluate the "goodness" of our concept embeddings on measuring entity semantic relatedness as an intrinsic evaluation. Entity relatedness has been recently used to model entity coherence in many named entity disambiguation systems.

4.1.1 Dataset

We use a benchmark dataset created by Ceccarelli et al. (2013) from the CoNLL 2003 data. As in previous studies (Yamada et al., 2016;

¹<http://dumps.wikimedia.org/enwiki/>

Method	nDCG@1	nDCG@5	nDCG@10	MAP
WLM (Huang et al., 2015)	0.54	0.52	0.55	0.48
Yamada et al. (2016)	0.59	0.56	0.59	0.52
3C	0.53	0.50	0.52	0.46
CRC	0.63	0.59	0.61	0.55

Table 1: Intrinsic evaluation on measuring entity semantic relatedness. The CRC model gives the best results outperforming all other models.

Top-level	Low-level
Sport	Hockey, Baseball, Autos, Motorcycles
Politics	Guns, Mideast, Misc
Religion	Christian, Atheism, Misc

Table 2: 20NG category mappings

Huang et al., 2015), we model measuring entity relatedness as a ranking problem. We use the test split of the dataset to create 3,314 queries. Each query has a query entity and ~ 91 response entities labeled as related or unrelated. The quality is measured by the ability of the system to rank related entities on top of unrelated ones.

4.1.2 Compared Systems

We compare our models with two prior methods:

1. [Yamada et al. \(2016\)](#) who used the skip-gram model to learn embeddings of words and entities jointly. The authors also utilized Wikipedia link graph to better model entity-entity relatedness.
2. [Witten and Milne \(2008\)](#) who proposed Wikipedia Link-based Measure (WLM) as a simple mechanism for modeling the semantic relatedness between Wikipedia concepts. The authors utilized Wikipedia link structure under the assumption that related concepts would have similar incoming links.

4.1.3 Results

We report Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (nDCG) scores as commonly used measures for evaluating the ranking quality. Table 1 shows the performance of our CRC and 3C models compared to previous models. As we can see, the 3C model performs poorly on this task compared to prior models. On the other hand, our CRC model outperforms all the other methods by 2-4% in terms of nDCG and by 3% percent in terms of MAP.

4.2 Dataless Classification

[Chang et al. \(2008\)](#) proposed dataless classification as a learning protocol to perform text categorization without the need for labeled data to train a classifier. Given only label names and few descriptive keywords of each label, classification is performed on the fly by mapping each label into a BOC representation using ESA. Likewise, each data instance is mapped into the same BOC semantic space and assigned to the most similar label using a proper similarity measure such as cosine. Formally, given a set of n labels $L = \{l_1, \dots, l_n\}$, a text document d , a BOC mapping model f , and a similarity function Sim , then d is assigned to the i th label l_i such that $l_i = \arg \max_i Sim(f(l_i), f(d))$. We evaluate the effectiveness of our concept embedding models on the dataless classification task as an extrinsic evaluation. We demonstrate through empirical results the efficiency and effectiveness of our proposed BOC densification scheme in obtaining better classification results compared to the original sparse BOC representation.

4.2.1 Dataset

We use the 20 Newsgroups dataset (20NG) ([Lang, 1995](#)) which is commonly used for benchmarking text classification algorithms. The dataset contains 20 categories each has ~ 1000 news posts. We obtained the BOC representations using ESA from [Song and Roth \(2014\)](#) who utilized a Wikipedia index containing pages with 100+ words and 5+ outgoing links to create ESA mappings of 500 dimensions for both the categories and news posts of the 20NG. We designed two types of classification tasks: 1) fine-grained classification involving closely related classes such as *Hockey* vs. *Baseball*, *Autos* vs. *Motorcycles*, and *Guns* vs. *Mideast* vs. *Misc*, and 2) coarse-grained classification involving top-level categories such as *Sport* vs. *Politics* and *Sport* vs. *Religion*. The top-level categories are created by combining instances of the

Method	Hockey x Baseball		Autos x Motorcycles		Guns x Mideast x Misc	
ESA	94.60	@425	72.70	@325	70.00	@500
3C (equal)	94.60	@20	-	-	70.33	@60
CRC (equal)	94.60	@60	73.10	@4	70.00	@7
WE_{max}	86.85	@65	76.15	@375	72.20	@300
WE_{hung}	95.20	@325	73.75	@300	71.70	@275
3C (best)	95.10	@125	69.70	@7	72.47	@250
CRC (best)	95.65	@425	79.20	@14	73.40	@70

Table 3: Extrinsic evaluation on dataless classification of fine-grained classes measured in micro-averaged F1 along with # of dimensions at which corresponding performance is achieved.

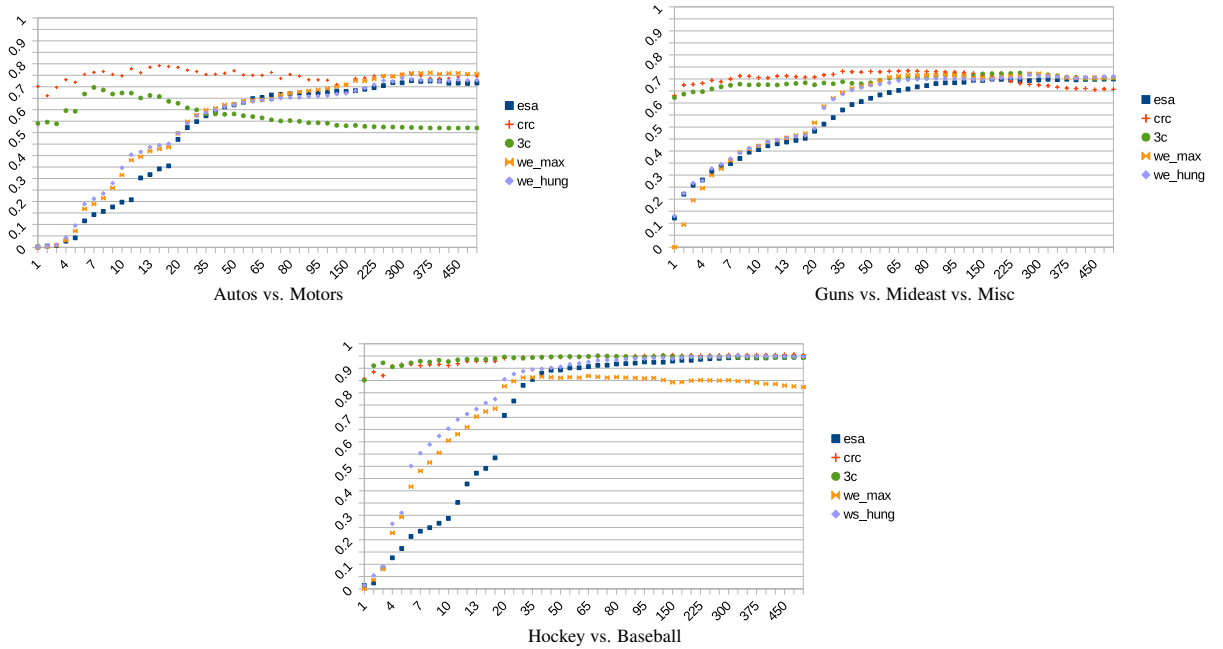


Figure 1: micro-averaged F1 scores of fine-grained classes when varying the # of BOC dimensions.

fine-grained categories as shown in Table 2.

4.2.2 Compared Systems

We compare our models with three prior methods:

1. **ESA** which computes the cosine similarity using the sparse BOC vectors.
2. **WE_{max} & WE_{hung}** which were proposed by [Song and Roth \(2015\)](#) for BOC densification using embeddings obtained from Word2Vec. As the authors reported, we fix the minimum similarity threshold to 0.85. WE_{max} finds the best match for each concept, while WE_{hung} utilizes the Hungarian algorithm to find the best concept-concept alignment on one-to-one basis. Both mechanisms have polynomial time complexity.

4.2.3 Results

Table 3 presents the results of fine-grained dataless classification measured in micro-averaged F1. As we can notice, ESA achieves its peak performance with a few hundred dimensions of the sparse BOC vector. Using our densification mechanism, both the CRC & 3C models achieve equal performance to ESA at much less dimensions. Densification using the CRC model embeddings gives the best F1 scores on the three tasks. Interestingly, the CRC model improves the F1 score by $\sim 7\%$ using only 14 concepts on *Autos vs. Motorcycles*, and by $\sim 3\%$ using 70 concepts on *Guns vs. Mideast vs. Misc*. The 3C model, still performs better than ESA on 2 out of the 3 tasks. Both WE_{max} and WE_{hung} improve the performance over ESA but

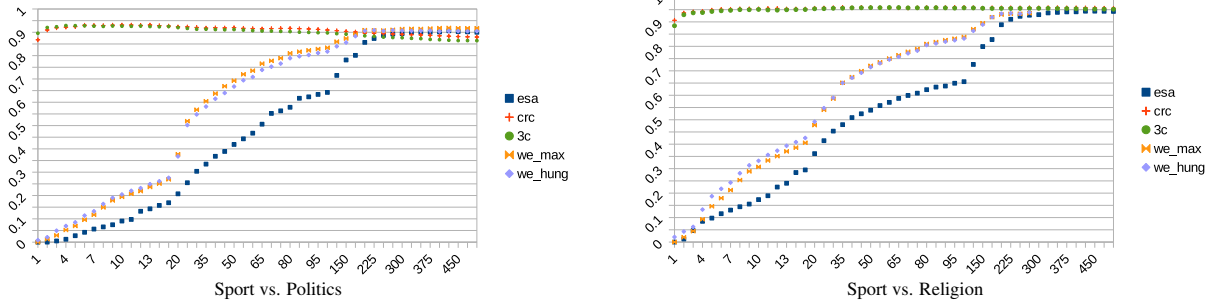


Figure 2: micro-averaged F1 scores of coarse-grained classes when varying the # of BOC dimensions.

Method	Sport x Politics		Sport x Religion	
ESA	90.63	@425	94.39	@450
3C (equal)	92.04	@2	95.11	@6
CRC (equal)	90.99	@2	94.81	@5
WE_{max}	91.89	@425	93.99	@425
WE_{hung}	90.89	@275	94.16	@450
3C (best)	92.89	@4	95.86	@60
CRC (best)	93.12	@13	95.91	@95

Table 4: Extrinsic evaluation on dataless classification of coarse-grained classes measured in micro-averaged F1 along with # of dimensions at which corresponding performance is achieved.

not as our CRC model.

In order to better illustrate the robustness of our densification mechanism when varying the # of BOC dimensions, we measured F1 scores of each task as a function of the # of BOC dimensions used for densification. As we see in Figure 1, with *one* concept we can achieve high F1 scores compared to ESA which achieves zero or very low F1. Moreover, near-peak performance is achievable with the top 50 or less dimensions. We can also notice that, as we increase the # of dimensions, both WE_{max} and WE_{hung} densification methods have the same undesired monotonic pattern like ESA. Actually, the imposed threshold by these methods does not allow for full dense representation of the BOC vector and therefore at low dimensions we still see low overall F1 score. Our proposed densification mechanisms besides their low cost, produce fully densified representations allowing good similarities at low dimensions.

Results of coarse-grained classification are presented in Table 4. Classification at the top level is easier than the fine-grained level. Nevertheless,

as with fine-grained classification, ESA still peaks with a few hundred dimensions of the sparse BOC vector. Both the CRC & 3C models achieve equal performance to ESA at very few dimensions (≤ 6). Densification using the CRC model embeddings still performs the best on both tasks. Interestingly, the 3C model gives very close F1 scores to the CRC model at less dimensions (@4 with *Sport vs. Politics*, and @60 with *Sport vs. Religion*) indicating its competitive advantage when computational cost is a decisive criteria. The 3C model, still performs better than ESA, WE_{max} , and WE_{hung} on both tasks.

Figure 2 shows F1 scores of coarse-grained classification when varying the # of BOC dimensions used for densification. The same pattern of achieving near-peak performance at very few dimensions recur with the CRC & 3C models. ESA using the sparse BOC vectors achieves low F1 up until few hundred dimensions are considered. Even with the costly WE_{max} and WE_{hung} densifications, performance sometimes decreases.

5 Conclusion

In this paper we proposed two models for learning concept embeddings based on the skip-gram model. We also proposed an efficient and effective mechanism for BOC densification which outperformed the prior proposed densification schemes on dataless classification. Unlike these prior densification mechanisms, our method scales linearly with the # of the BOC dimensions. In addition, we demonstrated through the results how this efficient mechanism allows generating high quality dense BOC vectors from few concepts alleviating the need of obtaining hundreds of concepts when generating the concept vector.

References

- Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. 2013. Learning relatedness measures for entity linking. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, pages 139–148.
- Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *AAAI*, volume 2, pages 830–835.
- Ofer Egozi, Shaul Markovitch, and Evgeniy Gabrilovich. 2011. Concept-based information retrieval using explicit semantic analysis. *ACM Transactions on Information Systems (TOIS)* 29(2):8.
- Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, pages 1606–1611.
- Zellig S Harris. 1954. Distributional structure. *Word*.
- Zhiting Hu, Poyao Huang, Yuntian Deng, Yingkai Gao, and Eric P Xing. 2015. Entity hierarchy embedding. In *Proceedings of The 53rd Annual Meeting of the Association for Computational Linguistics*.
- Hongzhao Huang, Larry Heck, and Heng Ji. 2015. Leveraging deep neural networks and knowledge graphs for entity disambiguation. *arXiv preprint arXiv:1504.07678*.
- Dongwoo Kim, Haixun Wang, and Alice H Oh. 2013. Context-dependent conceptualization. In *IJCAI*, pages 2330–2336.
- Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th international conference on machine learning*, pages 331–339.
- Yue Zhang Li, Ronghuo Zheng, Tian Tian, Zhiting Hu, Rahul Iyer, and Katia Sycara. 2016. Joint embedding of hierarchical categories and entities for concept categorization and dataless classification. *arXiv preprint arXiv:1607.07956*.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Christos H Papadimitriou and Kenneth Steiglitz. 1982. *Combinatorial optimization: algorithms and complexity*. Courier Corporation.
- Haoruo Peng, Yangqiu Song, and Dan Roth. 2016. Event detection and co-reference with minimal supervision. *EMNLP*.
- Walid Shalaby and Wlodek Zadrozny. 2015. Measuring semantic relatedness using mined semantic analysis. *arXiv preprint arXiv:1512.03465*.
- Yangqiu Song and Dan Roth. 2014. On dataless hierarchical text classification. In *AAAI*, pages 1579–1585.
- Yangqiu Song and Dan Roth. 2015. Unsupervised sparse vector densification for short text similarity. In *Proceedings of NAACL*.
- Yangqiu Song, Shusen Wang, and Haixun Wang. 2015. Open domain short text conceptualization: A generative+ descriptive modeling approach. In *IJCAI*, pages 3820–3826.
- Ian Witten and David Milne. 2008. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy*, AAAI Press, Chicago, USA, pages 25–30.
- Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. ACM, pages 481–492.
- Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. *arXiv preprint arXiv:1601.01343*.